

## **REMARKS/ARGUMENTS**

Claims 1, 4-8, 12, 15-18, and 21-24 are pending in the present application. Claims 12, 15-18, and 21-23 were canceled; claims 1 and 24 were amended; and no claims were added. No new matter has been added by the amendments to the claims. Support for the amendments to the claims may be found in the Specification in Figures 2, 9, 24, and 25 and their associated text. Reconsideration of the claims is respectfully requested.

Applicants have amended some claims and canceled others. Applicants do not concede that the subject matter encompassed by the earlier presented claims is not patentable over the art cited by the Examiner. Applicants canceled and amended claims in this response solely to facilitate expeditious prosecution of this application. Applicants traverse all rejections and respectfully reserve the right to pursue the earlier-presented claims, and additional claims, in one or more continuing applications.

### **I. 35 U.S.C. § 103, Obviousness**

The Final Office Action has rejected claims 1, 4-8, 12, 15-18, and 21-24 under 35 U.S.C. § 103 as being unpatentable over *Damron*, System and Method for Prefetching for Pointer Linked Data Structures, U.S. Patent No. 6,782,454, August 24, 2004 (hereinafter "*Damron*") in view of *Hooker*, Microprocessor with Repeat Prefetch Instruction, U.S. Patent Application Publication No. 2003/0191900, October 9, 2003 (hereinafter "*Hooker*"). This rejection is respectfully traversed.

Regarding this rejection, the Final Office Action states:

**As per claims 1 and 18**, *Damron* discloses a method in a data processing system for providing hardware assistance to prefetch data during execution of code by a processor in the data processing system, the method comprising:

responsive to loading of an instruction in the code into a cache, determining, by a processor unit, whether metadata for a prefetch is present for the instruction, wherein the metadata comprises a prefetch indicator that is associated with the instruction by one of being placed in the instruction or stored in a shadow memory (col. 4, lines 58-61; col. 5, lines 17-24; Fig. 1, element 175; Fig. 3, element 220); *It should be noted that computer program product in claims 18-23 executes the exact same functions as the methods in claims 1 and 4-6. Therefore, any references that teach claims 1-6 also teach the corresponding claims 18 and 21-23. It should also be noted that the "prefetch request" is analogous to the "instruction", the "prefetch engine" is analogous to the "processing unit", and the "starting address of a node (to be prefetched), an offset value, and a termination value" all in combination are analogous to the "metadata." Lastly, it should be noted that the "starting address of a node (to be prefetched)" and the "offset value", which are placed in the prefetch request, are analogous to the "prefetch indicator" being placed in the instruction.*

responsive to determination of the metadata being present for the instruction, determining whether data is to be prefetched (col. 5, lines 27-35; Fig. 3, element 230); It

should be noted that "*determining whether a termination condition has been satisfied*" is analogous to "*determining whether data is to be prefetched*." Data is prefetched until the termination condition is met.

and responsive to a determination that data is to be prefetched, prefetching data, from within a data structure using the metadata, into the cache in the processor (col. 5, lines 24-26 and 35-42; Fig. 3, elements 225 and 235). *It should be noted that responsive to the termination condition not being met, data is prefetched.*

Damron does not expressly disclose wherein the step of determining whether data is to be prefetched comprises one of determining whether a number of outstanding cache misses is less than a first threshold, and determining whether a number of cache lines chosen to be replaced is greater than a second threshold;

and wherein the step of prefetching comprises one of prefetching the data responsive to determining that the number of outstanding cache misses is less than the first threshold, and prefetching the data responsive to determining that the number of cache lines chosen to be replaced is greater than the second threshold.

Hooker discloses wherein the step of determining whether data is to be prefetched comprises one of determining whether a number of outstanding cache misses is less than a first threshold, and determining whether a number of cache lines chosen to be replaced is greater than a second threshold (paragraph 0069; Fig. 5, element 536); *It should be noted that the "response buffers" are analogous to the "cache lines."*

and wherein the step of prefetching comprises one of prefetching the data responsive to determining that the number of outstanding cache misses is less than the first threshold, and prefetching the data responsive to determining that the number of cache lines chosen to be replaced is greater than the second threshold (paragraph 0070; Fig. 5, element 538).

Damron and Hooker are analogous art because they are from the same field of endeavor, that being prefetching memory systems.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to implement Hooker's threshold based prefetch method within Damron's prefetch engine because, advantageously, by not prefetching cache lines if not enough free response buffers exist, the efficiency of a microprocessor is potentially increased. Therefore, it would have been obvious to combine Damron and Hooker for the benefit of obtaining the invention as specified in claims 1 and 18.

Final Office Action dated February 11, 2008, pp. 3-5 (emphasis in original).

The Examiner bears the burden of establishing a *prima facie* case of obviousness based on prior art when rejecting claims under 35 U.S.C. § 103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). The prior art reference (or references when combined) must teach or suggest all the claim limitations. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). In determining obviousness, the scope and content of the prior art are... determined; differences between the prior art and the claims at issue are... ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background, the obviousness or non-obviousness of the subject matter is determined. *Graham v. John Deere Co.*, 383 U.S. 1 (1966). "Often, it will be necessary for a court to look to interrelated teachings of multiple patents; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by a person having ordinary skill in the art, all in order to

determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue.” *KSR Int’l. Co. v. Teleflex, Inc.*, No. 04-1350 (U.S. Apr. 30, 2007). “*Rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.*” *Id.* (citing *In re Kahn*, 441 F.3d 977, 988 (CA Fed. 2006)).”

A *prima facie* obviousness rejection cannot be stated because neither *Damron* nor *Hooker*, either alone or in combination, teaches or suggests all of the features of claims 1 and 24. Currently amended independent claim 1, which is representative of claim 24 in regards to similarly recited subject matter, recites:

1. A method in a data processing system for providing hardware assistance to prefetch data during execution of code by a processor in the data processing system, the method comprising:
  - responsive to loading of an instruction in the code into a cache, determining, by a processor unit, whether metadata for a prefetch is present for the instruction, wherein the metadata comprises a prefetch indicator that is associated with the instruction and stored in a prefetch indicator field of a page table;
  - storing a first threshold value and a second threshold value in a first threshold field and a second threshold field, respectively, of the page table in association with the prefetch field and a plurality of counter fields, wherein the first threshold value specifies a maximum counter value in an associated counter field and wherein the second threshold value specifies a minimum counter value in an associated counter field;
  - responsive to a determination of the metadata being present for the instruction, determining whether data is to be prefetched, wherein the step of determining whether data is to be prefetched comprises determining whether a first counter value in an associated counter field representing a number of outstanding cache misses is less than the first threshold value in the first threshold field, and determining whether a second counter value in an associated counter field representing a number of cache lines chosen to be replaced is greater than the second threshold value in the second threshold field; and
  - responsive to a determination that data is to be prefetched, prefetching data, from within a data structure using the metadata, into the cache in the processor, wherein the step of prefetching comprises prefetching the data responsive to determining that the first counter value in an associated counter field representing the number of outstanding cache misses is less than the first threshold value in the first threshold field, and responsive to a determination that the second counter value in an associated counter field representing the number of cache lines chosen to be replaced is greater than the second threshold value in the second threshold field.

The combination of *Damron* in view of *Hooker* fails to teach or suggest the features of “wherein the metadata comprises a prefetch indicator that is associated with the instruction and stored in a prefetch indicator field of a page table,” “storing a first threshold value and a second threshold value in a first threshold field and a second threshold field, respectively, of the page table in association with the prefetch indicator field and a plurality of counter fields, wherein the first threshold value specifies a maximum counter value in an associated counter field and wherein the second threshold value specifies a minimum value in an associated counter field,” and “responsive to a determination of the metadata being present for

the instruction, determining whether data is to be prefetched, wherein the step of determining whether data is to be prefetched comprises determining whether a first counter value in an associated counter field representing a number of outstanding cache misses is less than the first threshold value in the associated first threshold field, and determining whether a second counter value in an associated counter field representing a number of cache lines chosen to be replaced is greater than the second threshold value in the associated second threshold field.”

Neither Damron nor Hooker teaches or suggests “wherein the metadata comprises a prefetch indicator that is associated with the instruction and stored in a prefetch indicator field of a page table.” Neither *Damron* nor *Hooker* teach, suggest, or even mention storing prefetch indicators in a prefetch indicator field of a page table.

Further, neither *Damron* nor *Hooker* teaches or suggests “storing a first threshold value and a second threshold value in a first threshold field and a second threshold field, respectively, of the page table in association with the prefetch indicator field and a plurality of counter fields, wherein the first threshold value specifies a maximum counter value in an associated counter field and wherein the second threshold value specifies a minimum value in an associated counter field.” Neither *Damron* nor *Hooker* teach, suggest, or even mention the use of page tables comprising a prefetch indicator field, a plurality of counter fields and two threshold value fields, wherein the first threshold value specifies a maximum counter value in an associated counter field and wherein the second threshold value specifies a minimum value in an associated counter field. At best, *Hooker* teaches a single threshold value stored in a threshold register 182 of Figure 1. Thus, *Hooker* does not teach or suggest a threshold value that specifies a maximum counter value in an associated counter field and a threshold value that specifies a minimum value in an associated counter field or “storing a first threshold value and a second threshold value in a first threshold field and a second threshold field, respectively, of the page table in association with the prefetch indicator field and a plurality of counter fields, wherein the first threshold value specifies a maximum counter value in an associated counter field and wherein the second threshold value specifies a minimum value in an associated counter field.”

Additionally, neither *Damron* nor *Hooker* teach or suggest “responsive to a determination of the metadata being present for the instruction, determining whether data is to be prefetched, wherein the step of determining whether data is to be prefetched comprises determining whether a first counter value in an associated counter field representing a number of outstanding cache misses is less than the first threshold value in the first threshold field, and determining whether a second counter value in an associated counter field representing a number of cache lines chosen to be replaced is greater than the second threshold value in the second threshold field.”

The Final Office Action admits that *Damron* does not expressly disclose “wherein the step of determining whether data is to be prefetched comprises determining whether a number of outstanding cache misses is less than a first threshold, and determining whether a number of cache lines chosen to be replaced is greater than a second threshold.” Thus, *Damron* does not teach or suggest the feature of “responsive to a determination of the metadata being present for the instruction, determining whether data is to be prefetched, wherein the step of determining whether data is to be prefetched comprises determining whether a first counter value in an associated counter field representing a number of outstanding cache misses is less than the first threshold value in the first threshold field, and determining whether a second counter value in an associated counter field representing a number of cache lines chosen to be replaced is greater than the second threshold value in the second threshold field.”

*Hooker* also fails to teach “wherein the step of determining whether data is to be prefetched comprises determining whether a first counter value in an associated counter field representing a number of outstanding cache misses is less than the first threshold value in the first threshold field, and determining whether a second counter value in an associated counter field representing a number of cache lines chosen to be replaced is greater than the second threshold value in the second threshold field.” At best, *Hooker* teaches prefetching a cache line if a number of free response buffers is greater than threshold. However, the threshold taught by *Hooker* is a number of free response buffers, which is different than either a first counter value in an associated counter field representing a number of outstanding cache misses is less than the first threshold value in the first threshold field or determining whether a second counter value in an associated counter field representing a number of cache lines chosen to be replaced is greater than the second threshold value in the second threshold field. Further, claim 1 recites a threshold value stored in a threshold field of a page table, which *Hooker* does not teach or suggest.

The Final Office Action states that response buffers are analogous to cache lines. (See Final Office Action, mailed February 11, 2008, page 4) However, Applicants respectfully disagree. In *Hooker*, the threshold being stored in the threshold register 162 of Figure 1 is a number of free threshold buffers that includes a number of buffers to insure that higher priority instructions have available response buffers for allocation. (See *Hooker*, paragraphs [0050]-[0053] and [0069]-[0073]) In contradistinction, claim 1 recites a first counter value in an associated counter field representing a number of outstanding cache misses is less than the first threshold value in the first threshold field and a second counter value in an associated counter field representing a number of cache lines chosen to be replaced is greater than the second threshold value in the second threshold field. Thus, as recited in claim 1, the number of cache lines to be replaced has no relationship to a number of buffers being available to be allocated to receive the cache lines. The determination of whether a number of cache lines to be replaced is greater than a

threshold value is made independent of and irregardless to the number of response buffers being available for allocation. Thus the threshold taught by *Hooker* is not the same as “a second counter value in an associated counter field representing a number of cache lines chosen to be replaced is greater than the second threshold value in the second threshold field,” as recited in claim 1.

Even assuming *arguendo* that *Hooker* could be construed as teaching “a second counter value in an associated counter field representing a number of cache lines chosen to be replaced is greater than the second threshold value in the associated second threshold field,” *Hooker* still does not teach that a determination of whether to prefetch is based “determining whether a first counter value in an associated counter field representing a number of outstanding cache misses is less than the first threshold value in the first threshold field, and determining whether a second counter value in an associated counter field representing a number of cache lines chosen to be replaced is greater than the second threshold value in the second threshold field.” *Hooker* does not teach such a two-prong test for determining whether to prefetch a cache line. Rather, *Hooker* merely teaches prefetching if a number of response buffers available to be allocated to receive the data is greater than a threshold.

Furthermore, as neither *Damron* nor *Hooker* teaches or suggests the feature of “responsive to a determination of the metadata being present for the instruction, determining whether data is to be prefetched, wherein the step of determining whether data is to be prefetched comprises determining whether a first counter value in an associated counter field representing a number of outstanding cache misses is less than the first threshold value in the first threshold field, and determining whether a second counter value in an associated counter field representing a number of cache lines chosen to be replaced is greater than the second threshold value in the second threshold field,” logically, neither *Damron* nor *Hooker* can teach or suggest the feature of “responsive to a determination that data is to be prefetched, prefetching data, from within a data structure using the metadata, into the cache in the processor, wherein the step of prefetching comprises prefetching the data responsive to determining that the first counter value in an associated counter field representing the number of outstanding cache misses is less than the first threshold value in the first threshold field, and responsive to a determination that the second counter value in an associated counter field representing the number of cache lines chosen to be replaced is greater than the second threshold value in the second threshold field.” That is, neither *Damron* nor *Hooker* can teach or suggest performing an action in response to making a determination that they did not teach making.

Therefore, for at least the reasons set forth above, Applicants submit that the combination of *Damron* in view of *Hooker* fails to render claim 1 obvious. Applicants respectfully submit that the Office Action fails to state a *prima facie* case of obviousness in regards to claim 1. Thus, Applicants submit that claim 1 is in condition for allowance over the cited references. Furthermore, as claim 1 is representative

of independent claim 24, the same distinction between claim 1 and the cited references also apply to claim 24. Additionally, as claims 4-8 depend from independent claim 1, Applicants submit that claims 4-8 are also in condition for allowance at least by virtue of their depending from an allowable base claim. Claims 12, 15-18, and 21-23 have been canceled.

Therefore, the rejection of claims 1, 4-8, 12, 15-18, and 21-24 under 35 U.S.C. § 103 has been overcome.

## **II. Conclusion**

It is respectfully urged that the subject application is patentable over the cited reference and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: May 12, 2008

Respectfully submitted,

/Gerald H. Glanzman/

Gerald H. Glanzman  
Reg. No. 25,035  
Yee & Associates, P.C.  
P.O. Box 802333  
Dallas, TX 75380  
(972) 385-8777  
Attorney for Applicants

GG/blj